

**Tecnologie Web T (9 cfu)**  
**Prova d'Esame "Pre-Natalizia" – 23 Dicembre 2021 – Versione C**

**Tempo a disposizione: 180 minuti**

---

La soluzione comprende la consegna elettronica dei seguenti file:

<b>C1.zip</b>	file zip contenente il sorgente java/class e pagine Web per punto 1
<b>C2.zip</b>	file zip contenente il sorgente java/class e pagine Web per punto 2
<b>C3.zip</b>	file zip contenente pagine Web e codice React.js per punto 3

**Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, descrittori, risorse statiche o dinamiche, codice Java e relativi .class, ecc.) e NON dell'intero progetto.**

**N.B.** Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), ben distribuiti sui 3 esercizi, ovvero in ciascuno dei tre esercizi si deve raggiungere una valutazione almeno quasi sufficiente.

---

**ESERCIZIO 1 (11 punti)**

Si realizzi una applicazione Web per **l'elaborazione di contenuti testuali lato servitore**, basandosi principalmente sulle tecnologie Java Servlet e JSP.

L'applicazione Web deve permettere ai soli utenti autenticati di accedere ad una pagina HTML in cui inserire un testo la cui lunghezza deve essere compresa fra 50 e 5000 caratteri. All'inserimento del carattere speciale "\$", la pagina deve automaticamente richiedere al servitore di effettuare l'elaborazione del testo inserito. In particolare, il testo deve essere passato a un primo componente server-side che deve eliminare tutti i caratteri non alfabetici eventualmente presenti; il risultato di questa elaborazione deve essere passato a un secondo componente server-side che dovrà trasformare tutte le lettere maiuscole in minuscole; infine il terzo componente dovrà contare quante parole cominciano per "B". Tutti i componenti utilizzati server-side devono essere servlet. Il risultato finale del conteggio, unitamente all'elenco delle parole che cominciano per "B", deve essere inviato al browser Web in formato JSON.

**Monitoraggio:** l'applicazione deve anche tenere traccia, relativamente agli ultimi 30 giorni, di quante volte è stata invocata da ogni cliente, di quante sessioni ogni cliente ha avuto attive, e di quante invocazioni complessive vi sono state da parte di tutti i clienti. Qualora il numero delle invocazioni complessive superi 10000 nell'arco di 30 giorni, allora tutti gli utenti devono essere notificati il prima possibile con una pagina Web con l'indicazione "Superato il numero massimo di invocazioni complessive!".

**Gestione:** si preveda inoltre che un amministratore di sistema, previa autenticazione, possa avere accesso a una pagina **admin.jsp** in cui sono elencate tutte le sessioni correntemente attive e in cui si possa forzare la terminazione della sessione di un qualsiasi utente.

**Tecnologie Web T (9 cfu)**  
**Prova d'Esame "Pre-Natalizia" – 23 Dicembre 2021 – Versione C**

**ESERCIZIO 2 (11 punti)**

Si realizzi un **sito Web di informazioni** (tipo <http://www.corriere.it>) in grado di fare push dal server al cliente degli ultimi aggiornamenti di notizie. L'applicazione Web deve essere basata principalmente su tecnologie Javascript, AJAX e servlet.

In particolare, al primo accesso l'applicazione Web deve inviare all'utente le informazioni relative alle notizie accadute nelle ultime 24 ore. Dopo questo primo accesso, **seguito un paradigma di interazione long polling**, il server dovrà essere in grado di operare dei push degli aggiornamenti delle informazioni sfruttando la tecnologia Javascript+AJAX lato cliente. Le news inviate dal server al cliente devono essere trasferite **in formato XML**.

Infine, si realizzi una *estensione* (versione 2) dell'applicazione descritta sopra che **invalidi la sessione utente e lo costringa a ri-iniziare l'interazione da capo dopo 5 minuti di inattività (nessuna richiesta esplicita inviata) del cliente stesso**. La terminazione di sessione deve essere comunque comunicata al cliente tramite una opportuna pagina Web che segnali "Sessione terminata dal server" non appena possibile dopo la scadenza dei 5 minuti indicata sopra.

---

**ESERCIZIO 3 (11 punti)**

Si realizzi in React un'applicazione Web lato client che simula il gioco della **battaglia navale single-player**. L'applicazione dovrà eseguire interamente sul browser senza interagire con alcun server remoto.

L'interfaccia dell'applicazione sarà composta dalle seguenti sezioni:

- **Sezione Configurazione.** In questa sezione sarà presente un bottone per il caricamento da oggetto JSON di un campo di battaglia già configurato. L'oggetto JSON sarà una matrice 6X6 in cui, per ciascuna riga, saranno collocate esattamente due navi, aventi rispettivamente dimensioni 1 cella e 2 celle. Acquisito il campo, potrà cominciare la battaglia;
- **Sezione Campo di Battaglia.** In questa sezione, in funzione della configurazione caricata, dovrà essere visualizzata una matrice di celle. L'utente avrà a disposizione 10 colpi da "sparare" su altrettante celle del campo. Un colpo si intende sparato al momento del click del mouse sulla cella. Se il colpo affonda una nave di dimensione 1, oppure affonda definitivamente una nave di dimensioni 2 già danneggiata precedentemente, sulle rispettive celle dovrà apparire il simbolo "\*". Se il colpo danneggia una nave di dimensione 2 senza affondarla, sulla cella dovrà apparire il simbolo "+". Infine, se il colpo va a vuoto sulla cella corrispondente dovrà apparire il simbolo "X";
- **Sezione Punteggio.** In questa sezione, occorrerà riportare per ciascun colpo il relativo esito ("affondata", "colpita" o "mancata") e il punteggio guadagnato (10, 5 e 0 rispettivamente). Infine, occorrerà visualizzare il punteggio accumulato, aggiornato dopo ogni colpo da parte del giocatore.